REMARKS

Claims 1-12, 14-17, 19-22 and 24-35 are now pending in this application. The Examiner has objected to claim 1 for minor informalities. Claim 1, 14, 19 and depending claims stand rejected under 35 U.S.C. § 101 as being non-statutory subject matter. Claims 1, 3-12, 14, 16-17, 20-22, 24 and 26-35 stand rejected under 35 U.S.C. § 102(e) as being allegedly anticipated by United States Patent 6,922,708 ("Sedlar"). Claims 2, 15, 19 and 25 stand rejected under 35 U.S.C. § 103(a) as being allegedly anticipated by Sedlar in view of United States Patent 7,035,874 ("Reed").

Applicants respectfully traverse. Claims 1, 14, 19 and 24 have been amended.

Objection to Claim 1

Claim 1 has been amended to cure the objection raised by the Examiner and should now be in condition for allowance.

Rejection of Claims 1, 14 and 19 Under 35 U.S.C. § 101

Claims 1, 14 and 19 stand rejected under 35 U.S.C. § 101. Specifically, the Examiner asserts that these claims recite a method that fails to pass the machine or transformation test. Thus, the rejections of claims 28-29 under 35 U.S.C. § 101 should be withdrawn. Applicants asset that these claims, as amended, now perform transformations such as recited in the steps *extracting a real file system path corresponding to the item using the database engine function; and using the real file system path to perform an operation on the item.*

Rejection of Claims 1-12

Claims 1 and 3-12 stand rejected under 35 U.S.C. § 102(e) as being allegedly anticipated by Sedlar. Claim 2 stands rejected under 35 U.S.C. § 103(a) as allegedly anticipated by Sedlar in view of Reed. Claim 1 as amended recites a method for executing a transaction comprising at least one query language statement and at least one file system statement, each statement relating to a user defined type ("UDT") associated with a database server, comprising *storing at least one field relating to the UDT in a relational database table, wherein at least one field is a filestream field wherein data for each filestream field is* *stored in a respective file separate from the relational database table* and receiving each of the at least one file system statements, wherein each statement comprises a call to open a first item and at least one of a call to read from the first item and to write to the first item, and a call to close the first item.

Claim 1 as amended further recites *upon detecting a storage platform path name in a file system statement, forwarding the file system statement to an agent, wherein the agent performs a call to the storage platform by passing a path name corresponding to the first item to the storage platform, identifying the first item based upon the path name associated with the first item, passing a database engine function that returns a file system path name corresponding to the first item to the database server, performing a table look-up for the UDT associated with the first item on the database server, extracting a real file system path corresponding to the first item using the database engine function,* and *using the real file system path to perform an operation on the first item by passing the real file system path back to the agent, wherein the agent then interacts with the file system to cause execution of file system statement*.

Support for this amendment may be found on pages 6-9 of the specification. Briefly summarizing this discussion, the application relates to a method and system for unify the query language transaction and locking model with the file system sharing model. Transaction support is provided for file system operations so they may be executed in the context of a transaction. File system operations such as Win 32 operations may be executed in the context of a transaction. In addition, the sharing model for file system operations is unified with the transaction and locking model for query languages such as T-SQL providing an overall model for manipulating items that contain filestream fields in a storage platform.

Referring to Fig. 1 of the application, instances of storage platform items may be stored in a relational database table. The table may include two columns including an ID column and UDT column. Each ID may provide a unique identifier for a corresponding UDT, which corresponds to an instance of a type item. Fields of UDTs may be designated as filestream fields meaning that data for that field is stored separately from the table itself.

An exemplary storage platform environment is shown in FIG. 2. A client application may manipulate items in a data store directly through a storage platform or via "out of band" access through a file system API. If the client application manipulates items directly through the storage platform, such manipulation may be performed using a query language such as T-SQL. If, on the other hand, the client application manipulates items via "out of band" access through the file system API, such manipulation may be performed using a file system API such as Win32.

The client application may access the storage platform directly using storage platform methods to initiate a query on the data store. In particular, the storage platform may translate statements provided into a query in T-SQL or another query language and submit the translated statements to the data store. The data store may then execute the query against corresponding instances of the UDT and return stored values for each matching instance. The UDT objects may be wrapped and subsequently returned to the application.

The storage platform may also enable "out of band" access to filestream fields in items via the file system API. The client application may initiate "out of band" access by making a call via an interface on the storage platform and passing a path name to the storage platform that identifies the requested data based on the identity of the corresponding field in the instance of a persisted UDT. When the file system API receives a command from the client application that include the storage platform path name, the file system API may recognize it as such and then forward it to a FS agent. The FS agent may then issue a call to the storage platform, passing the storage platform path name of the item field. The storage platform may then identify the item and field from the storage platform path name and then pass this information to a database engine. The storage platform may pass a database engine function that returns the file system pathname for a filestream field of a UDT object that has been stored separately from the database store.

The database engine may respond to the request by performing a table look-up in the table in which the UDT object that is the subject of the request is stored. The database engine may position to the correct row of the table and then to serialized fragments of the UDT object within that row. For the filestream field in question, the database engine may extract from its corresponding fragment the real file system path to the file in which the data for that field is stored. The database engine may send the real path back to the storage platform, which in turn passes the file system pack to the FS agent, which in turn calls the file system API to open the file passing the real file system path to the request. The file system API then obtains a handle to the file and passes it back to the client application.

Sedlar relates to a method for performing file system statements. Calls to perform a plurality of file system operations are received through a file system interface. The plurality of file system operations are performed as a single transaction by performing the steps of : if all file system operations of the plurality of file system operations are completed without failure, making permanent all changes made by the plurality of operations and if any of the plurality of file system operations fail, then undoing all changes made by all the plurality of file operations. Contrary to the Examiners assertions, Sedlar fails to teach or suggest storing a filestream field separately from a relational database table as recited in claim 1. The indices cited by the Examiner are not files, which store the substance of a data file as defined in the application. Further Sedlar fails to teach or suggest unifying the transaction and model with the file sharing model as argued in the previous amendment and recited in the uncited portions of the claim.

Nonetheless, despite the fact that Sedlar fails to teach or suggestion these limitations, that reference nowhere teaches the specific manner of processing file system statements recited in amended claim 1, namely upon detecting a storage platform path name in a file system statement, forwarding the file system statement to an agent, wherein the agent performs a call to the storage platform by passing a path name corresponding to the first item to the storage platform, identifying the first item based upon the path name associated with the first item, passing a database engine function that returns a file system path name corresponding to the first item to the database server, performing a table look-up for the UDT associated with the first item on the database server, extracting a real file system path corresponding to the first item using the database engine function, and using the real file system path to perform an operation on the first item by passing the real file system path back to the agent, wherein the agent then interacts with the file system to cause execution of file system statement.

Thus, as Sedlar fails to teach or suggest the cited claim limitations, claim 1 as amended should be allowed. Claims 2-12 depend from and therefore include all the limitations of claim 1. And, as Reed fails to cure the deficiencies of Sedlar, claims 2-12 should also be allowed.

Rejection Of Claims 14-17

Claim 14 as amended recites limitations similar to claim 1 including *upon detecting a storage platform path name in the file system statement, performing a call to a storage platform by passing a path name corresponding to the item to the storage platform, identifying the item based upon the path name associated with the item, passing a database engine function that returns a file system path name corresponding to the item to a database server, performing a table look-up for the UDT associated with the item on the database server, extracting a real file system path corresponding to the item using the database engine function and using the real file system path to perform an operation on the item.* Thus, for at least the reasons stated with respect to claim 1, claim 14 should also be allowed.

Claims 15-17 depend from and therefore include all the limitations of claim 14. Thus, for at least the reasons stated with respect to claim 14, claims 15-17 should also be allowed.

Rejection Of Claims 19-22

Claim 19 as amended recites limitations similar to claim 1 and 14 including *upon detecting a storage platform path name in the file system statement, performing a call to a storage platform by passing a path name corresponding to the item to the storage platform, identifying the item based upon the path name associated with the item, passing a database engine function that returns a file system path name corresponding to the item to a database server, performing a table look-up for the UDT associated with the item on the database server, extracting a real file system path corresponding to the item using the database engine function and using the real file system path to perform an operation on the item.* Thus, for at least the reasons stated with respect to claims 1 and 14, claim 19 should also be allowed.

Claims 20-22 depend from and therefore include all the limitations of claim 19. Thus, for at least the reasons stated with respect to claim 19, claims 20-22 should also be allowed.

Rejection Of Claims 24-35

Claim 24 as amended recites limitations similar to claim 1, 14 and 19 including *a file system, wherein the file system is adapted to upon detecting a storage platform path name in the file system statement, performing a call to the storage platform by passing a path name corresponding to the item to the storage platform, identifying the item based upon the path* *name associated with the item, passing a database engine function that returns a file system path name corresponding to the item to the relational data engine, wherein the relational data engine is adapted to, upon receiving a database engine function from the file system, perform a table look-up for an associated UDT, extract a real file system path corresponding to the item using the database engine function, pass the real file system path to perform an operation on the item to the file system to cause the file system to perform an operation on the item.*

Thus, for at least the reasons stated with respect to claims 1, 14 and 19, claim 24 should also be allowed. Claims 25-35 depend from and therefore include all the limitations of claim 24. Thus, for at least the reasons stated with respect to claim 24 claims 25-35 should also be allowed.

CONCLUSION

In view of the above amendments and remarks, applicant respectfully submits that the present invention is in condition for allowance. Reconsideration of the application is respectfully requested.

Date: March 4, 2009

/Kenneth R. Eiferman/
Kenneth R. Eiferman
Registration No. 51,647

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439